

通過テスト4 解答

4

クラスとファイル

1. E

【解説】クラスメソッドは最初から `class` ファイルの中に存在している。しかし、その `Class` ファイルの中に `main` メソッドがあるときそれは実行可能なファイルである。`main` メソッドがない場合は実行可能ではないので **A** は正しい言い方ではない。異なる `OS` 上でも同じ `class` ファイルを使用できるのが `Java` の特徴であるので **B** は誤り。`java` ファイルはソースプログラムファイルなので例外なくコンパイルしなくては実行できない。したがって **C** は誤り。ひとつのファイルに複数のクラスを書いてコンパイルすると、複数の `class` ファイルが作成されるので、**D** は誤り。ひとつの `java` ファイルに複数のクラスを書いた場合、ファイル名と同じ名前のクラスだけを `public` アクセスにできるので、**E** は正しい。

2. D

【解説】`Cat` クラスはデフォルトアクセスなので同じパッケージ内のクラスからしか利用できない。

3. A

【解説】

32-33 頁の「`static` と静的メンバ」を読み返してください。`static` のついた変数を **クラス変数** と呼び、一般の変数は **インスタンス変数** といいます。インスタンス変数がオブジェクトの中にコピーされて、各オブジェクトごとに存在するの比べて

- ・クラス変数はクラス情報 (`class` ファイル) としてメモリ上にロードされて存在します
- ・クラス変数は個々のオブジェクトにコピーされません
- ・クラス変数は同じクラスのメソッドやコンストラクタから共有変数として利用されます

クラス変数をアクセスするには、同じクラス内であれば変数名を書くだけで構いませんが、他のクラスからアクセスする際にはクラス名を付ける必要があります。

`n` ----- クラス内での利用

`Dog.n` ----- クラス外からの利用 (クラス内でこのように書いても誤りではない)

問題では、同じクラス内ですから、単に `n` と書くか、あるいは `Dog.n` とかくのが正しい書き

方です。ところが、`Dog dog = new Dog();` としてオブジェクトを作成し、それを利用して

`dog.n`

という使い方をしています。これは、インスタンス変数を利用する時の書き方で、この場合は、「クラスメンバである変数 `n`」という意味になります。確かに `n` は `Dog` クラスのメンバですからこのように書いてもコンパイルエラーにはなりません。ですから正解は **A** です。しかし、クラス変数としては、正しい使い方ではなく、`Eclipse` では警告がでます。

本来は以下のように書くのが正しい書き方です。

```
public class Dog {
    static int n=1;
    public static void main(String[] args){
        Dog dog = new Dog();
        System.out.println(n); // あるは Dog.n
    }
}
```

なお、この問題と直接の関係はありあせんが、問題としてしばしば出題されるので、アクセス修飾子と `import` 文との関係にも注目しておきましょう。`Dog` クラスをパッケージが異なるクラスから利用するには `import` 文が必要ですが、この問題では `n` には `public` が付いていません。そのため `Dog.n` を直接には利用できないことになります。

4. ① `public class Cat`

② `String n`

③ `say()`

④ `import pk10.Cat;`